Electronic Version 1.1

Stylesheet Version v1.1.1

Description

METHOD FOR COMPRESSING XML DOCUMENTS INTO VALID XML DOCUMENTS

TECHNICAL FIELD

[0001] This document relates generally to compression algorithms for data files and in particular to compressing extensible markup language (XML) documents.

BACKGROUND

[0002] The extensible markup language (XML) is a language that is written in the standardized general markup language (SGML). SGML is an international standard meta-language for text markup applications (ISO 8879). XML is a human-readable, text-based language making it easy to use. Partly because XML is written in an international standard and partly because of its ease of use, XML is widely used in a variety of applications. Another advantage is that XML files or documents explicitly flag the type of data contained in the documents by enclosing blocks of data with labels to declare the type of XML elements contained in a block. This makes XML documents data-type aware.

[0003] However, because it is human-readable and because it is data-type aware, XML can be a verbose language. Human-readable data files are larger compared to other formats (such as binary formats for example) and the data-type declarations expand the size of data files. Large XML files may cause problems in systems that are memory constrained or in communication systems having channels that are bandwidth limited.

SUMMARY

[0004] This document describes both devices and methods used to manage extensible markup language (XML) files or documents. One method example comprises compressing a first XML document into a binary stream, converting the binary stream into a compressed valid XML document, and associating at least one XML tag with the compressed valid XML document in order to identify the document as a compressed XML document.

[0005] One device example includes at least one processor, a network interface to communicate with the at least one processor and a network, and an XML document processing module. The XML document processing module includes a compression module to compress XML documents into compressed valid XML documents.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] FIG. 1 shows a block diagram of one embodiment of a method of managing XML documents.

[0007] FIG. 2 shows a block diagram of another embodiment of a method of managing XML documents.

[0008] FIG. 3 is block diagram illustrating portions of a network device operable to manage XML documents.

[0009] FIG. 4 is block diagram illustrating portions of another embodiment of a network device operable to manage XML documents.

[0010] FIG. 5 is a block diagram of portions of an embodiment of a system for managing XML documents.

[0011] FIG. 6 is an embodiment using an XML tag at the beginning and end of the file.

[0012] FIG. 7 is an original XML document configuration file.

[0013] FIG. 8 is the compressed version of the document.

DETAILED DESCRIPTION

[0014] In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

[0015] This document discusses, among other things, methods and devices for managing extensible markup language (XML) files or documents. Because XML is widely used, many applications that use XML would benefit from reducing the size of XML documents. This is especially true where the applications are memory constrained such as in embedded systems. Application files that are reduced in size would allow the files to be stored using less memory. Applications that include bandwidth limited communication systems would also benefit from reducing the size of XML files. These applications include those that are slow, such as a slow serial line, or those that experience a large amount of communication traffic such as a wide area network (WAN). These applications would benefit from minimizing traffic by minimizing the amount of data transferred.

[0016] To manage the size of large XML documents, the documents are compressed. In contrast to typical compression methods however, documents compressed under the methods of the present application remain valid XML documents. A valid XML document is a document that is well formed and has an associated document-type declaration. This allows the compressed valid XML document to be recognized and accessed by applications that process XML documents.

[0017]

FIG. 1 shows a block diagram 100 of one embodiment of a method of managing XML documents. The method includes reducing the size of the document by compressing it. At 110, an XML document is compressed into a binary stream.

Because XML documents are plain text, they are very redundant. Any compression method that results in good compression ratios on redundant text streams may be used. A 70% compression ratio is a typical good compression ratio. In one embodiment, the compression method is a deflate compression algorithm, such as RFC 1951 for example.

[0018] At 120, the binary stream is converted into a compressed valid XML document. To accomplish this, the binary stream is expanded back into text. This is necessary because valid XML documents cannot contain binary data. Preferably, the binary stream is expanded using base-64 encoding, but any encoding mechanism that has characteristics similar to base-64 encoding may be used. A mechanism with similar characteristics refers to an encoding mechanism that takes binary bytes of data and converts them into printable characters in the American Standard Code for Information Interchange (ASCII) standard. For example, UUencode has similar characteristics.

[0019] If base-64 encoding is not used, the resulting text must be searched through after encoding. Any characters that are present in the text that would cause the resulting document to be invalid XML must be converted. For example, use of the '<' or '>' characters would result in an invalid XML document, and the characters must be replaced with the standard XML replacement text '&lt' and '&gt', respectively. Base-64 does not require this replacement step because the resulting characters only consist of upper and lower case A-Z, numerals 0-9, '+', '/', and '=' which are valid XML characters. The result of using binary to ASCII encoding is an expansion of the binary stream back into a text file with an expansion ratio of about 33%.

[0020] The net result of the compression and expansion is approximately a 2.5 to 1 compression from the original XML document. At 130, at least one XML tag is associated with the compressed valid XML document in order to identify the document as a compressed XML document. An example of an embodiment using an XML tag at the beginning and end of the file is shown in FIG. 6.

[0021] In a specific example, the XML document is a configuration file used to configure a remote device. A description of using XML documents to configure remote devices is included in co-pending U.S. Patent Application 10/873,051, entitled "DEVICE SERVER ACCESS USING A DATA TYPE AWARE MARK-UP LANGUAGE," which is incorporated herein by reference. An example original XML document configuration file is shown in FIG. 7. The compressed version of the document of FIG. 7 is shown in FIG. 8.

[0022] The compressed document looks random because of compression and expansion with encoding. However, the compressed document is a valid XML document.

[0023] Note that due to space limitations, the original XML document is small, about 440 bytes. This results in the compressed document in the example being larger than the original XML document. Because of the way compression algorithms work, the algorithms only compress well when the original file is large, e.g., greater than 4000 bytes. For smaller documents, the amount of memory used or the amount of time spent in sending the document is not an issue and the method to manage the XML documents may not be needed.

[0024] Because the compressed document is valid XML, any application that can read XML documents will recognize the document and can access its contents. To use the document, the application must decompress the document to return the document to its original format. This involves reversing the compression and encoding process. Therefore, a further embodiment of the method 100 for managing XML documents includes reconverting the compressed valid XML document into a binary stream, and decompressing the binary stream to obtain the first XML document. In one embodiment, reconverting into binary includes reverse base-64 encoding, and decompressing includes running a reverse deflate algorithm on the interim binary stream. If a mechanism other than base-64 encoding is used, the replaced characters must be reconverted from their XML replacements.

[0025] FIG. 2 shows a block diagram of an embodiment of a method 200 that includes compressing and encoding a first XML document and then transferring a compressed valid XML document over a network. The first XML document is any generic document, such as a status message for example. At 210 a first XML document is compressed into a binary stream. At 220, the binary stream is converted into a compressed valid XML document. The compressing and encoding are accomplished by any of the methods discussed previously. At 230, at least one XML tag is associated with the compressed valid XML document that identifies the document as a compressed XML document. At 240, the compressed valid XML document is transferred over a network to a receiving device. At 250, the transferred document is recognized as a compressed valid XML document. According to some embodiments, a master device such as a master processor generates the compressed valid XML document and initiates the transfer to a remote device that recognizes the XML document from the at least one XML identifying tag. In an example of one such embodiment, the first XML document includes a configuration file. In another example, the first XML document includes a status message. In yet another example, the first XML document includes a command message. According to other embodiments, a remote device generates the compressed valid XML document and initiates the transfer. At 260, the compressed valid XML document is reconverted into a binary stream. At 270, the binary stream is decompressed to obtain the first XML document. A receiving device is then able to process the XML document.

[0026] In yet another embodiment, transferring the compressed valid XML document over a network includes transferring the compressed valid XML document over a serial communications network, such as a network that uses an RS232 protocol or an

Ethernet network. In yet another embodiment, transferring the compressed XML document over a network includes transferring the compressed valid XML document over a wireless network, such as a wireless local area network (WLAN) or a mobile phone network. In a further embodiment, transferring the compressed XML document over a network includes transferring the compressed valid XML document over the internet. In other embodiments, one or a combination of the several method embodiments are provided on a computer readable medium such as a diskette or CD ROM.

[0027] FIG. 3 is block diagram illustrating portions of a network device 300 operable to manage XML documents. The network device 300 includes at least one processor 310, a network interface 320 to communicate with the at least one processor 310 and a network 330, and an XML document processing module 340.

[0028] The XML document processing module 340 includes a compression module 350 to generate compressed valid XML documents. In one embodiment, the XML document processing module 350 includes a deflate compression algorithm. In another embodiment, the XML document processing module 350 includes a binary to ASCII text encoding algorithm. In one such embodiment, the binary to ASCII text encoding algorithm includes a base-64 encoding algorithm.

[0029] In some embodiments, the network device 300 is an embedded device server operable to manage a remote device using XML documents. In another embodiment, the network interface 320 includes a serial port. In another embodiment, the network interface 320 includes a web interface. In another embodiment, the network 330 is a wireless network. In one such embodiment, the network device 300 is included in a cell phone. In another embodiment, the network 330 is a wireless local area network (WLAN) and the network device 300 is included in a WLAN computer card.

[0030] FIG. 4 is a block diagram of portions of an embodiment of a network device 400 operable to recognize a compressed valid XML document and reverse the compression process. The network device 400 includes at least one processor 410, a network interface 420 to communicate with the at least one processor 410 and a network 430, and an XML document processing module 440 that includes a compression module 450. To reverse the compression process, the network device 400 includes a decompression module to 460 decompress compressed valid XML documents. In one embodiment, the decompression module 460 includes a re-conversion algorithm to reconvert a compressed valid XML document into a binary stream and a reverse deflate algorithm to convert the interim binary stream into an XML document.

[0031] FIG. 5 is a block diagram of portions of an embodiment of a system 500 for communicating XML documents. The system 500 includes a communication network 530 and at least first and second network devices 505A-B to communicate over the network 530. Each network device 505A-B includes at least one processor 510A-B, a network interface 520A-B to communicate with the at least one processor 510A-B and the network 530, and an XML document processing module 540A-B. An XML document processing module 540A-B includes a compression module 550A-B to compress XML documents into compressed valid XML documents and a decompression module 560B to decompress compressed valid XML documents.

[0032] In one embodiment, a first network device 505A is operable to transfer a status message over the network 530 as a compressed valid XML document to a second network device 505B. In another embodiment, a first network device 505A is an embedded device server operable to receive a device configuration file as a compressed valid XML document over the network 530 and decompress the document. According to other embodiments, the network 530 is a serial communication network. In other embodiments, the network 530 is a wireless communication network.

[0033] The accompanying drawings that form a part hereof, show by way of illustration, and not of limitation, specific embodiments in which the subject matter may be practiced. The embodiments illustrated are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed herein. Other embodiments may be utilized and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. This Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[0034] Such embodiments of the inventive subject matter may be referred to herein, individually and/or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any single invention or inventive concept if more than one is in fact disclosed. Thus, although specific embodiments have been illustrated and described herein, it should be appreciated that any arrangement calculated to achieve the same purpose may be substituted for the specific embodiments shown. This disclosure is intended to cover any and all adaptations or variations of various embodiments. Combinations of the above embodiments, and other embodiments not specifically described herein, will be apparent to those of skill in the art upon reviewing the above description.

[0035] The Abstract of the Disclosure is provided to comply with 37 C.F.R. §1.72(b), requiring an abstract that will allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, it can be seen that various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separate embodiment.